BE Data pour ISA

Notes post-session 1 (oct 2018)

TODO : mettre ces notes dans un fichier séparé

On est arrivé après 3h, j'ai tout juste au tracé de la température intérieure du capteur Delta Dore (et les étudiants peut être aussi). Points aberrants: NaN ok, mais pas remplissage. Rééchantillonage et synchronisation pas fait. Wide vs narrow pas présenté (trop abstrait également).

À la fin, ils avaient décroché (fatigue, découragement ?).

Bonne accroche des étudiants sur les tracés de la température à Rennes et calcul de la moyenne.

Bases théoriques manquantes :

- logical indexing (étudiants et profs inclus !!)
- un fichier Word n'est pas un fichier texte. Un CSV est un fichier texte.

Problèmes techniques:

- (générique : raccourci Matlab manquant pour bcp dont poste prof)
- spécifique BE : la lecture du log capteur avec readtable donne une colonne 'type' de type cell array, alors que import Tool donne directement
- Bug sujet : Le lien de documentation Import Tool pointe vers https://fr.mathworks.com/help/matlab/ref/importtool-app.html qui donne la commande `uuimport` qui est un Import Wizard un peu différent (plus ancien : <=2006). C'est un bug de la doc → vérifier en R2018 si c'est tjs le cas, et sinon, marquer « ne pas faire uiimport »

Pour 2019:

R2018a améliore `retime` : on peut spécifier plus librement le pas de temps (plus précis que 'minutely', 'hourly'…)

Accélérer le démarrage (TODO : faire des diapos) et mettre les questions théoriques plus tard, car c'est trop abstrait pour la pluspart. Résumer le blabla sur les formats de fichier.

Les discussions théoriques doivent venir après la pratique.

Ajouter plus de fonction appelables dans le sujet pour permettre l'autonomie.

Expliciter le pourquoi du choix des trois fichiers : leur différences et leur caractère représentatif.

	Météo Rennes	**Log capteur**	**Fiche présence**
Date	4 colonnes numériques	2 colonnes texte (date est parsée autom., le temps plus compliqué)	1 colonne texte
Structure	Wide	Narrow (empilée)	
Données aberrantes	Oui		

Données catégorielles		Oui	Oui
Échantillonage	Régulier	Irrégulier	Irrégulier

~~Simplifier les dates du log Delta Dore pour gagner du temps ?~~

- MAL: non, c'est une difficulté à surmonter qu'ils doivent rencontrer.
- PH : important de savoir ce qu'est la chaîne de format de datetime.

Cadrage initial avant séance (sept 2018)

Format :

- « BE » = une séance de 2 × 1h30
- Sur poste Matlab
- Co animation Marie Anne & Pierre

Problématiques à aborder

1. Propreté du code final

Faire une fonction pour charger les données et faire les manip bas-niveau

Généricité : pas de constantes numériques codés en dur (e.g. « for i=1:100 »)

2. Format : fichier, organisation

Fichier : xls, csv, largeur fixe

Représentation propre en mémoire

- structures (data.t, data.x), Timeseries

3. Données manquantes & aberrantes

- Aberrantes (hors gamme)
- Manquante, NaN, NA
- TODO : recherche de vocabulaire (« structurellement manquante »)

4. Horodatage, rééchantillonage

Problèmes :

- Échantillonage irrégulier
- Asynchronie entre des séries :
  - non synchrones (horloges différentes)
  - périodes d'échantillonage différentes

Besoins :

- changer l'échantillonage :

    - interpoler : linéaire vs. blocage. ⚠ *ça dépend du type de donnée (e.g. variables discrètes)*
    - sous-échantillonner : brutal, avec préfiltrage, ou bien agrégation (moyennage) ?
- resynchroniser des séries

5. Se Définir un but

Qu'est-ce qu'on veut chercher dans les données :

- vérifier la cohérence
- visualisation (quoique ce soit vague)
- modélisation (e.g. identification)

6. Visualisation

Types de représentations graphiques:

- tracé temporel (e.g. pb des variables discrète → pas d'interpolation, ou bien couleur d'arrière plan)
- distribution : histogramme
- relation entre séries : nuage de point

Problématiques intéressantes mais HS

Lien avec Simulink (to/from workspace)

Activités en séance

1 à 2 exemples de jeu de données à analyser :

- capteur Delta Dore
- Barrage de la Rance

Comment : **Autonomie**, découverte par soi-même (avoir du temps pour chercher, en particulier lire des docs). Peut-être éviter de les lancer dans des recherches Google (trop grande dispersion).

- Greg Wilson (cf. après) recommande cependant de commencer par des **exemples résolus** (live coding)

Évaluation

Tester la fonction de chargement de données par les pairs?

À creuser...

Notions qu'on voudrait aborder

Chargement de fichier CSV ou autres

- cf. liste des options possibles dans l'aide Matlab "Data Import and Export" https://fr.mathworks.com/help/matlab/data-import-and-export.html
- load : fichier Matlab
- outil interactif Import Tool, MAIS à qui il faut demander de **générer une fonction**
- readtable, avec ses options

Texte : CSV, txt, ...	Spreasheet (Excel) : .xls(x)	Matlab .mat
csvread dlmread textscan readtable (2013b) → table	xlsread readtable	load

Structure de données en mémoire

- plusieurs vecteurs
- `table`

Données manquantes & aberrantes

Comment les détecter? Comment les remplacer ou pas? → pas de réponse toute faite (dépend du cas, de l'application)

Fonction ismissing. Valeur NaN (pour les nombres).

Et puis parfois elles sont justes probablement aberrantes (ex: données de direction "scotchées" à 0°)

Question d'étape :

- Calculer la moyenne sur l'année à Rennes
- Combien de données manquantes?

Représenter les données temporelles

Représentation interne obscure : type `datetime` et son compagnon `duration`.

Et 3 représentations concrètes possibles

- Un réel, compté depuis un t0 implicite/explicite
- Représentations texte
- Représentation en triplet / sextuplet de nombres

Représentation texte ↔ parseur

Structures de donnes en mémoire (bis)

- timetable (la spécialisation de table)
- timeseries, spécifique au temps (structure autour de 2 vecteurs : t et x + métadonnées)

Un peu de visualisation

on arrive naturellement (saute aux yeux, av bon fichier) aux valeurs manquantes (NaN et/ou absentes) & aberrantes (-9999)

Question d'étape :

- Visualiser un extrait temporel.
- Échantillonage régulier?

Traiter les données (temporelles)

Remplir les trous, (éventuellement ?)

rééchantillonage (interpolation, au plus proche…). retime(), ou resample()

- Exemple : données Delta Dore température → on voit bien la différence sur une rééchantillonage à la minute (perte de l'effet de quantification de la mesure avec 'linear').
- Données de présence : il faut faire un blocage 'previous' (données discrètes)

sous-échantillonage/agréger : retime(… 'mean')

Importance de synchroniser les données entre elles pour faire des analyses

Objectif final: On termine par des jolis nuages de point

- CO2 ~ Température intérieur (pure Delta Dore)
- en dernier : CO2 ~ Présence

Idée : utiliser les données Delta Dore plus tard (commencer avec Météo Rennes), car il faut les désempiler.

Plan de la séance (draft sept 18)

Accueil. Présentation des objectifs de la séance

Activités 1, 2, 3, 4

Important : avoir un "take home message" pour conclure chaque activité

Charger un CSV, fichier texte

Dates : Attention String vs. internal format `datetime` (opérations parse / format)

bonne pratique : utiliser les types dédiés

formats de fichiers: [carte à trou]

- binaire (Excel .xls and .xlsx)
- texte
  - CSV : séparateurs
  - fixed width

Exo : décrire les fichiers abc1-4 puis les charger

bonne pratique : ouvrir dans un éditeur de texte

Structures : Table, Timetable, Timeseries

Variable catégorique (vs. quantitative, R) : efficacité du stockage

Table : Chaque colonne est une “variable”, chaque ligne…

narrow/stacked vs. **wide/unstacked** data format

Q : quel est l’intérête du format empilé pour le log DeltaDore?

Rééchantillonage

Delta Dore : OK (données non sync)

Q sur l’échantillonage:

- est-il régulier?
  - `isregular(timetable)`
  - Diagnostic : `plot diff(ts.time)` ou bien `diff(datetime)/minutes`
- y a-t-il des trous dans l’enregistrement (!= y a-t-il des valeurs manquantes)

functions `retime, synchronize`

- rééchantillonage régulier (implique un léger sur/sous échantillonage)
- Sous échantillonage, agrégation
- trouver des plages communes lorsqu’il y a plusieurs sources

Difficulté : comment obtenir l’interpolation linéaire lors d’un rééchantillonage, sans combler les gros trous de valeurs manquantes ?

- Activité : détecter des trous dans un enregistrement ?

Conclusion : l’analyste doit faire un choix

- soit on rebouche les trous (si “petits”) → interpolation linéaire, ou bien plus proche voisin
- soit on élimine toute la zone

Données manquantes/aberrante

Q : comment plot affiche-t-il les NA ?

Diabète : petits trous, aberrantes

DeltaDore : gros trous, ~~aberrantes~~

NOAA ISD : petits trous (aberrantes)

Un peu de visualisation, mais pas trop

Visualiser qq corrélations

- présence vs. Temp/CO2

Review of literature and existing courses

Data Carpentry workshops

https://datacarpentry.org/lessons/#ecology-workshop

- Data Organization in Spreadsheets
  - *plan à étoffer*
- Data Cleaning with OpenRefine
- Data Analysis and Visualization in Python
  - *plan à étoffer*

“Tidy Data” concept

From “Good enough practices in scientific computing” [1]

1. Data management

1. *Save the raw data.*
2. *Create the data you wish to see in the world.*
3. *Create analysis-friendly data* ("tidy" data).
4. *Record all the steps used to process data.*
5. *Anticipate the need to use multiple tables, and use a unique identifier for every record.*

From “Ten Simple Rules for Digital Data Storage” [2]

- Rule 5: Data Should Be Structured for Analysis

One such structure for data stores (“Codd’s 3rd normal form”)

- each variable as a column,
- each observation as a row
- each type of observational unit as a table

“Tidy data” article (quite long) [3]

Data Cleaning

OpenRefine tool [4], taught during Data Carpatentry workshops

Data Exploration

“What is Visualization?” video (Introduction to Data Exploration and Visualization, ASU) [5]

→ Visualization is not about pretty pictures, but about ???

Exploratory Data Analysis (EDA) concept

Data Vizualization

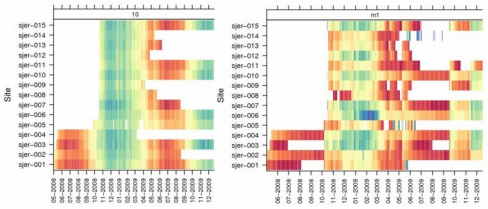
types of plots :

- pandas doc http://pandas.pydata.org/pandas-docs/stable/visualization.html
- Seaborn doc: https://seaborn.pydata.org/tutorial.html
  - Visualizing statistical relationships
  - Plo tting with categorical data
  - Visualizing the distribution of a dataset

Categorical time series

https://stats.stackexchange.com/questions/107571/help-creating-a-chart-to-show-categorical-data-over-time

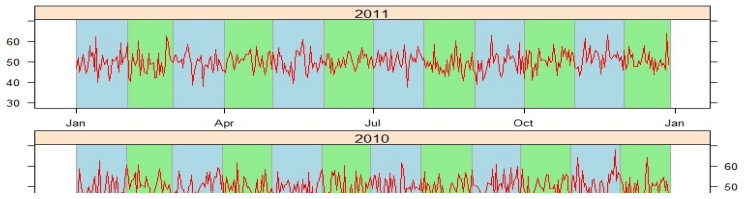
https://casoilresource.lawr.ucdavis.edu/blog/interesting-use-levelplot-time-series-data/



http://rgraphgallery.blogspot.com/2013/04/rg-trellis-plot-of-time-series-plot.html

https://www.ggplot2-exts.org/ggTimeSeries.html

- Calendar Heatmap



How to teach computing

Exercises in a computing session

Exercise Types http://teachtogether.tech/en/exercises/ ++ nice typology

After reading this chapter, you will be able to...

- *Describe four types of formative assessment exercises for programming classes.*
- *Describe two kinds of feedback on programming exercises that can be given by automated tools.*

What Kinds of Exercises Do You Use to Teach Programming? http://third-bit.com/2017/10/16/exercise-types.html

“Teaching Tech Together” book

“A Little Bit of Theory” chapter http://teachtogether.tech/en/theory/ (NOT READ YET)

Ten Simple Rules for Creating an Effective Lesson

Aug 18, 2018. I’m preparing a couple of talks based on Teaching Tech Together, and this seems like a useful topic to cover. Updated notes are below;

http://third-bit.com/2018/08/18/ten-simple-rules-for-creating-an-effective-lesson.html

6. Use Concreteness Fading

- PETE (**Problem, Explanation, Theory, Example**) goes from specific and tangible to more abstract
- What is the authentic problem that the lesson solves next?
- Explain a concrete solution
- Fill in the underlying theory
- Provide a second example so that learners will understand which parts generalize
- Authentic problem may be an end goal, or in later lessons, may arise out of a previous solution
- Build a usable mental model so that learners have somewhere to put knowledge, then correct the model as necessary
- E.g., ball-and-spring model in chemistry, evolution solely by descent, CPU-memory-disk model in computing

7. Design for Peer Instruction

8. Design Around Worked Examples

- Learners learn faster from worked examples than they do from solving problems on their own
    - They eventually need to do the latter, but step-by-step explanation of why and how helps more
- Live performances (music, programming, theorem proof) are effectively worked examples
    - The “PEE” in “PETE”

9. Show How to Detect, Diagnose, and Correct Common Mistakes

- One aspect of worked examples that’s important enough to deserve its own section
- Novices spend much of their time making mistakes and trying to fix them, because they’re novices
- Including DD&C in the lesson reduces frustration, which in turn accelerates learning
- Also helps solidify their mental model

References

[1] G. Wilson, J. Bryan, K. Cranston, J. Kitzes, L. Nederbragt, and T. K. Teal, “Good enough practices in scientific computing,” *PLOS Comput. Biol.*, vol. 13, no. 6, p. e1005510, Jun. 2017.

[2] E. M. Hart *et al.*, “Ten Simple Rules for Digital Data Storage,” *PLOS Comput. Biol.*, vol. 12, no. 10, p. e1005097, Oct. 2016.
[3] H. Wickham, “Tidy Data,” *J. Stat. Softw.*, vol. 59, no. 10, pp. 1–23, 2014.
[4] “OpenRefine (formerly Google Refine).” [Online]. Available: http://openrefine.org/. [Accessed: 31-Aug-2018].
[5] “What is Visualization? - Introduction to Data Exploration and Visualization,” *Coursera*. [Online]. Available: https://www.coursera.org/lecture/intro-to-data-exploration/what-is-visualization-LdIbK. [Accessed: 31-Aug-2018].